Monotone Submodular Function Maximization Problem and Greedy Algorithm

Takaya KOIZUMI

Mathematical Science, B4

Applied Mathematics and Physics informal seminor, 3rd

Contents

Contents

劣モジュラ関数

V を有限集合とし, V の冪集合を 2^V と表記する. また, $S \subset V$, $e \in V$ に対して $\Delta_f(e|S) := f(S \cup \{e\}) - f(S)$ とする.

Definition (劣モジュラ関数)

集合関数 $f : 2^V \to \mathbb{R}$ が,

$$\forall A, B \subset V, \forall e \in V, (A \subset B \text{ かつ } e \in V - B) \Longrightarrow \Delta_f(e|A) \geq \Delta_f(e|B)$$

を満たす時, f を劣モジュラ関数 *(submodular function)* と呼ぶ.

単調劣モジュラ関数の性質

Theorem (同値条件)

f を集合関数とする．この時

1. f は劣モジュラ関数である．
2. $\forall A, B \subset V, f(A \cap B) + f(A \cup B) \leq f(A) + f(B)$

は同値である．

Proposition

f, f_1, f_2 を単調劣モジュラ関数とし, $\alpha > 0$ とする．この時,

1. $f_1 + f_2$ は劣モジュラ関数.
2. αf は劣モジュラ関数.

が成立する．

単調劣モジュラ関数

Definition (単調性)

f を集合関数とする. f が単調 *(Monotone)* であるとは

$$\forall S, T \subset V, S \subset T \Longrightarrow f(S) \leq f(T)$$

を満たすことを言う.

Definition (単調劣モジュラ関数)

集合関数 f が劣モジュラかつ単調であるとき f を単調劣モジュラ関数 *(Monotone Submodular function)* と言う.

Contents

Monotone Submodular Function Maximization Problem

これ以降 $f : 2^V \to \mathbb{R}$ を非負の単調劣モジュラ関数とする.

個数制約単調劣モジュラ関数最大化問題

$k \in \mathbb{N}$ を $0 \leq k \leq |V|$ を満たす自然数とする.

$$\arg\max_{|S| \leq k} f(S)$$

を求める問題を個数制約単調劣モジュラ関数最大化問題と呼ぶ。

今回は, 個数制約単調劣モジュラ関数最大化問題のみを扱うため, 単に単調劣モジュラ関数最大化問題または, MSFMP と呼ぶことにする.

どうやって MSFMP を解くのか?

まず, 全探索を用いることを考える. 全探索とは, ありえる可能性全てを列挙するアルゴリズムであり, この方法を使えば, 確実に厳密な最小解をえることができる. コンピュータで実装する際は, bit 全探索というものを用いれば簡単に実装することができる.

```
int n = 10;
// {0, 1, ..., n-1} の部分集合の全探索
for (int bit = 0; bit < (1 << n); bit++) {
    vector<int> S;
    for (int i = 0; i < n; ++i) {
        if (bit & (1 << i)) {  // bitの右からi番目にbitがたっているか
            S.push_back(i);
        }
    }
```

Figure: C++による実装例

全探索の問題点

じゃあ, 全探索でよくね? となってしまうが, 全探索は現実的な方法ではない. なぜならば, 全探索ではありえる可能性全てを列挙しなければならないため, $|V|$ が大きい時に計算量が爆発的に多くなってしまうからである. 以下に $|V|$ ごとの大まかな実行時間を示す. (ちなみに地球が誕生したのは 46 億年前)

V のサイズ	実行時間 (s)
10	0.0082
30	30
40	30720 (約 8 時間)
50	31457280 (8738 時間, 約 1 年)
100	35417748621522339102720 (約 1×10^{17} 年)

Table: MacBook Pro 2017, 2.5 GHz デュアルコア Intel Core i7, 16GB での計測結果 (40, 50, 100 に関しては推定値)

Contents

第二の選択肢：貪欲法

では, 全探索を諦めて, 貪欲法 (Greedy Algorithm) で解くことを考える. 貪欲法とは, 簡単にいえばある状態においてその状態において最も良い解を選択するアルゴリズムである.

貪欲法の具体例

値段が 100, 200, 350, 400, 500 円の 5 つの商品がある. この中から 3 つの商品を選ぶ時, 最も値段が高くなるような選び方はどれか?
解：最初に 500, 次に 400, 最後に 350 円の品物を選べは良い.

貪欲法であれば, 先ほどのように $|V|$ が大きくても, 高速に解を求めることができる.

貪欲法の問題点

貪欲法には問題がないのかというと，そういうわけではない．貪欲法の問題点は，問題によっては最小・最大解が得られる保証が全くないことである．例えば，以下のナップザック問題 (Knapsack Problem) では一般に貪欲法では最小解をえることはできない．

Knapsack Problem

以下の表の中から容量の合計が 100 以下で価値の合計が最大によるような商品の選び方を見つけなさい．

価値	30	40	35	30	50
容量	30	50	20	15	50

(価値に関しての) 貪欲法だと左から 2, 5 番の商品が解として選択されるが，実際の解は左から 1,3,5 番目の商品を選んだ時である．

Contents

近似率

貪欲法が真の解が得られないことは前スライドで説明した．では，MSFMP に関して貪欲法はどれくらい良い解を得ることができるのだろうか？ この疑問に答えるために，以下で定義される近似率を調べることとする．

Definition (Approximation rate)

近似アルゴリズム $\mathcal{A}$ によって得られる近似解を $S_{\mathcal{A}}$, 真の解を S_{OPT} とする．

$$f(S_{\mathcal{A}}) \geq \alpha f(S_{OPT})$$

が成立する最大の $\alpha > 0$ を $\mathcal{A}$ の近似率と呼ぶ．

MSFMP に関しての貪欲法

Algorithm 1 Greedy Algorithm for MSFMP

Require: Monotone Submodular function $f \geq 0$

Require: Constraints number k

initialize a set S_0 by empty set.

for i in 1..k **do**

find $s_i \in \underset{e \in V}{\arg\max}\, \Delta_f(e \mid S_{i-1})$

$S_i = S_{i-1} \cup \{s_i\}$

end for

return S_k

これ以降, MSFMP の最適解を S_{OPT}, $\{S_i\}_{i=0}^k$ を貪欲法で得られる集合列とする．この時, アルゴリズムの設計から明らかに $\{S_i\}_{i=0}^k$ は単調増加であり, f の単調性から $\{f(S_i)\}_{i=0}^k$ も単調増加である．

証明のための準備 その1

Lemma (補題その1)

f を劣モジュラ関数とする．この時

$$\forall S, T \subset V, S \subset T \implies f(T) - f(S) \leq \sum_{e \in T-S} \Delta_f(e|S)$$

が成立する．

Proof.

$s = |T - S|$, $T - S = \{i_1, i_2, \ldots, i_s\}$ とする．$S_\ell = S_{\ell-1} \cup \{i_\ell\}$ とした時 (ただし $S_0 = S$), f の劣モジュラ性から,
$f(S_{\ell-1} \cup \{i_\ell\}) - f(S_{\ell-1}) \leq f(S \cup \{i_\ell\}) - f(S)$.
両辺の $\ell = 1, 2, \ldots, s$ の和を取れば，主張を得る． $\square$

証明のための準備 その 2

Lemma (補題その 2)

任意の $\ell \in \{0, 1, 2, \cdots, k-1\}$ に対して

$$\begin{aligned} f(S_{\ell+1}) - f(S_\ell) &\geq \frac{1}{|S_{OPT} - S_\ell|}(f(S_{OPT}) - f(S_\ell)) \\ &\geq \frac{1}{k}(f(S_{OPT}) - f(S_\ell)) \end{aligned}$$

が成立する.

Lemma (補題その 3)

$$\forall x \in \mathbb{R}, 1 - x \leq e^{-x}$$

補題その 2 の証明

Proof.

$\ell \in \{1, 2, \cdots, k\}$ を任意に取る．f の単調性，S_ℓ の定義，補題その 1 より

$$
\begin{aligned}
f(S_{OPT}) - f(S_{\ell-1}) &\le f(S_{OPT} \cup S_{\ell-1}) - f(S_{\ell-1}) \\
&\le \sum_{i \in S_{OPT} - S_{\ell-1}} (f(S_{\ell-1} \cup \{i\}) - f(S_{\ell-1})) \\
&\le |S_{OPT} - S_{\ell-1}| \max_{i \in S_{OPT} - S_{\ell-1}} (f(S_{\ell-1} \cup \{i\}) - f(S_{\ell-1})) \\
&\le |S_{OPT} - S_{\ell-1}| (f(S_\ell) - f(S_{\ell-1})) \\
&\le k(f(S_\ell) - f(S_{\ell-1})).
\end{aligned}
$$

□

貪欲法の近似率

Theorem (貪欲法の近似率, Nemhauser)

$$\forall \ell \in \{1, 2, \cdots, k\}, f(S_\ell) \geq \left(1 - e^{-\frac{\ell}{k}}\right) f(S_{OPT})$$

が成立する.

これから貪欲法の近似率が $1 - \frac{1}{e} \approx 0.62$ 以上であることがわかる.

定理の証明

Proof.

$\ell \in \{0,1,2,\cdots,k-1\}$ を任意にとり，各 $i \in \{0,1,2,\cdots,k-1\}$ に対して $\delta_i = f(S_{OPT}) - f(S_i)$ とする．すると補題その 2 より

$$\delta_{\ell+1} \leq \left(1 - \frac{1}{k}\right) \delta_\ell$$

である．この不等式と f の非負性より
$\delta_0 = f(S_{OPT}) - f(\emptyset) \leq f(S_{OPT})$ であることと，補題その 3 を用いれば

$$\delta_\ell \leq \left(1 - \frac{1}{k}\right)^\ell \delta_0 \leq e^{-\frac{\ell}{k}} f(S_{OPT})$$

が従う．δ_ℓ の定義より

$$f(S_\ell) \geq \left(1 - e^{-\frac{\ell}{k}}\right) f(S_{OPT})$$

□

References

[1] 劣モジュラ最適化と機械学習・河原 吉伸, 永野 清仁・2015